REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

[illegible]

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED
FINAL/01 APR 93 TO 31 MAR 94

4. TITLE AND SUBTITLE
STOCHASTIC LEARNING DYNAMICS AND NON-LINEAR
DIMENSION REDUCTION

5. FUNDING NUMBERS
2304/HS
F49620-93-1-0253

6. AUTHOR(S)
PROFESSOR TODD LEEN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
OREGON GRADUATE INSTITUTE OF SCIENCE & TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY
P.O.BOX 9100
PORTLAND, OR 97291-1000

8. PERFORMING ORGANIZATION
REPORT NUMBER
AFOSR-TR- 95 - 0211

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 DUNCAN AVE, SUTE B115
BOLLING AFB DC 20332-0001

10. SPONSORING/MONITORING
AGENCY REPORT NUMBER
F49620-93-1-0253

11. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

13. ABSTRACT (Maximum 200 words)

This reports cover research activity under the grant in place from April 1993 through March 1994. Our research addressed algorithms and theory for stochastic learning, non-linear extensions of principal component analysis (PCA) for dimension-reduction, network pruning, and methods to incorporate desired invariances into learning.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to stay within the lines to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank).

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract		PR - Project	
G - Grant		TA - Task	
PE - Program Element		WU - Work Unit Accession No.	

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

[illegible]

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

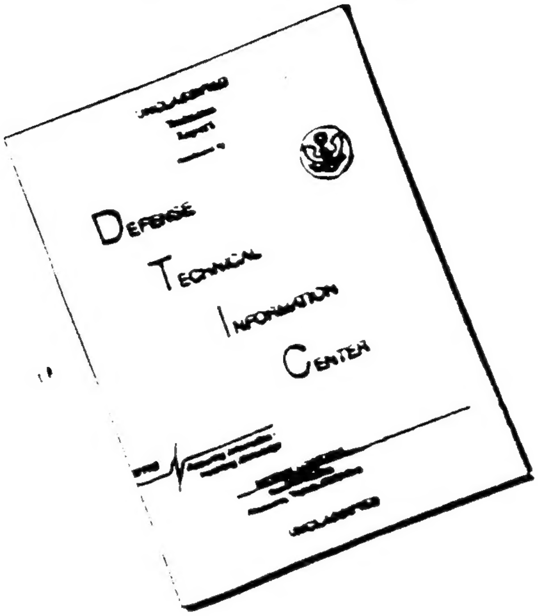
Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

DISCLAIMER NOTICE



Defense
Technical
Information
Center

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Stochastic Learning Dynamics and Non-Linear Dimension Reduction

Final Report for AFOSR Grant F49620-93-1-0253

Todd K. Leen, Associate Professor
Department of Computer Science and Engineering
Oregon Graduate Institute of Science & Technology
P.O. Box 91000
Portland, OR 97291-1000
tleen@cse.ogi.edu

October 19, 1994

Abstract

This report covers research activity under the grant in place from April 1993 through March 1994. Our research addressed algorithms and theory for stochastic learning, non-linear extensions of principal component analysis (PCA) for dimension-reduction, network pruning, and methods to incorporate desired invariances into learning.

In stochastic learning we extended and refined theoretical analysis developed prior to the grant period [1, 2, for example], focusing on asymptotic behavior in order to develop algorithms that improve late-time convergence rates. The resulting algorithms modify stochastic gradient methods by implicitly incorporating information about cost function curvature, without computing large Hessian matrices [3, 4].

Our algorithms for dimension reduction extend traditional PCA by developing *non-linear* data models. Our algorithms build locally linear models. The advantages over PCA are more accurate and compact data representations. The advantage over neural net approaches is that our techniques can be several orders of magnitude quicker to train. Under the present grant we improved the algorithms' accuracy, exercised them on a broader range of data (including speech and image data), and began to relate our algorithms to Gaussian-mixture models and use them for classification [5, 6].

Our work on network pruning introduced a fast algorithm for removing excess degrees of freedom. Like all regularization techniques the aim is to reduce model variance at the cost of bias to improve performance on out of sample data. Network pruning techniques typically use the Hessian to quantify the bias introduced by removing specific network weights. For large networks this is intractable and approximations (often poor) to the Hessian are adopted. Instead, we examine the correlation between node activities and use this, in conjunction with PCA, to estimate the bias introduced by removing degrees of freedom. This technique avoids large matrix calculations and is fast and effective [7].

Our most recent work establishes a correspondence between two methods for incorporating invariances into pattern recognition. Ideally pattern recognition machines provide constant output when the inputs are transformed under a group of desired invariances (e.g. translational and rotational invariance in machine vision). Two methods to achieve this (there are others) are i) enhancing the training data to include examples of inputs transformed by elements of the group, while leaving the corresponding targets unchanged, and ii) adding to the cost function a regularization term that penalizes changes in the output when the input is transformed under the group. Our work relates the two approaches, showing precisely the sense in which the regularized cost function approximates the result of adding transformed examples to the training data [8].

1 Objectives

Below we list the objectives for the research areas summarized in the abstract:

1.1 Stochastic Learning

- Refine the theoretical treatment to avoid the diffusion approximation of the earlier work. This has been accomplished by developing a perturbation expansion for the appropriate master equation and relating this to van Kampen's system size expansion [9].
- Extend the analysis to algorithms with learning rate annealing. To achieve convergence (in mean square or with probability one), stochastic gradient algorithms typically employ learning rate, or gain, schedules that behave asymptotically as μ_0/t with t the iteration number, and μ_0 the initial learning rate. For such schedules the rate at which the logarithm of the squared weight error decays appears to be bounded below by $1/t$. This optimal rate is achieved if $\mu_0 > \mu_{crit}$ where μ_{crit} is determined by the Hessian of the cost function at the minimum.

  We extended our analysis to such learning rate schedules and reproduced known results on asymptotic convergence rates and distributions.
- Extend the treatment of asymptotics to stochastic gradient descent with momentum. The motivation for this was to develop algorithms that insure the optimal convergence rate *without* knowledge of the Hessian. Intuitively, since "momentum" terms track previous updates, second differences of the cost function are implicitly contained in such algorithms. Using the analysis as a spring-board, we developed algorithms that incorporate an *adaptive* momentum coefficient and succeeds in obtaining optimal convergence rates.

1.2 Non-Linear Data Modeling

- Further develop our locally-linear dimension reduction algorithms to enhance the accuracy of the representations obtained.
- Exercise the algorithms on a broader range of data (both speech and image data) for comparison with PCA and neural-net-based approaches.

The results of these studies were published in [5].

- Relate the locally linear models to parametric distributions, in particular Gaussian mixture models, and extend their use from dimension reduction to classification. The results of this work will appear in [6].

1.3 Network Pruning

- Make use of the correlation in node activities to prune *effective* degrees of freedom from networks. This technique was motivated by the desire to prune models quickly, and without computation of large Hessian matrices.

  This work was published in [7] and was one of fewer than %6 of the submitted papers chosen for oral presentation at the 1993 Neural Information Processing Systems conference. The technique has been incorporated into algorithms for time-series prediction in the financial marketplace.

1.4 Invariances in Learning

- Develop the correspondence between data set enhancement, and regularization [10, for example], or "hints" [11, for example] as means to provide invariance in learning machines.

  This work has been accepted for publication in Neural Computation, and will be presented at the 1994 NIPS conference [8].

2 Quantitative Performance Measures

1. Publications in Journals – 2
   - (a) Leen, T.K., A Coordinate-Independent Center Manifold Reduction, *Physics Letters*, A-174, 89-93, 1993.
   - (b) Leen, T.K., Distortions and Regularization in Pattern Recognition, *Neural Computation*, to appear.
2. Publications in Conference Proceedings – 6
   - (a) Orr, G.B. and Leen, T.K.: Momentum and Optimal Stochastic Search, in *Proceedings of the 1993 Connectionist Models Summer School*, M.C. Mozer, P. Smolensky, D.S. Touretzky, J.L. Elman and A.S. Weigend (eds.), Erlbaum Associates, 1993.
   - (b) Leen, T.K. and Orr G.B.: Momentum and Optimal Stochastic Search, in J.D. Cowan, G. Tesauro and J. Alspector (eds.), *Advances in Neural Information Processing Systems, 6*, Morgan Kauffman Publishers, 1994.
   - (c) Leen, T.K. and Kambhatla, N.: Fast Non-Linear Dimension Reduction, in J.D. Cowan, G. Tesauro and J. Alspector (eds.), *Advances in Neural Information Processing Systems, 6*, Morgan Kauffman Publishers, 1994.
   - (d) Levin, A.U. and Leen, T.K.: Fast Pruning Using Principal Components, in J.D. Cowan, G. Tesauro and J. Alspector (eds.), *Advances in Neural Information Processing Systems, 6*. Morgan Kauffman Publishers, 1994. This paper was one of 6% of the submissions chosen for oral presentation at the 1993 NIPS conference.
   - (e) Kambhatla, N and Leen, T.K.: Classifying with Gaussian Mixtures, Clusters, and Subspaces, *Advances in Neural Information Processing Systems, 7*, to appear.
   - (f) Leen, T.K.: From Data Distributions to Regularization in Invariant Learning, *Advances in Neural Information Processing Systems, 7*, to appear.
3. Journal article in preparation:
   - (a) Leen, T.K., Orr, G.B, and Moody J.E.: Ensemble Theory of Stochastic Learning.
4. Books or book chapters – 0
5. Graduate students supported – Nanda Kambhatla and Genvieve Orr. Both are expected to complete the Ph.D. by June, 1995. (Partial support for the PI and students is from a grant from the Electric Power Research Institute. The AFOSR grant supported one graduate student full time.).
6. Postdoctoral associates supported – 0
7. External honors
   - (a) The PI served as theory program co-chair for the 1993 Neural Information Processing Systems conference.
   - (b) The PI is currently serving as Workshops Chair for the 1994 Neural Information Processing Systems conference.
   - (c) The PI was promoted to Associate Professor in July 1993.
   - (d) The PI was recently invited to join the editorial board of *Neural Computation*.

3 Accomplishments

We are engaged in research in two primary areas: stochastic learning, and non-linear data modeling and dimension reduction. Additional work on model pruning and learning invariances was supported under the grant and is discussed in this report. Our work is reported in the NIPS 1993 conference proceedings [4, 5, 7] and in the upcoming NIPS 1994 conference [6, 8]. Some of the work on stochastic learning also appears in the Proceedings of the 1993 Connectionist Model Summer School [3]. The work on network pruning was also presented at the NATO Workshop on Statistics and Neural Networks [12]. In addition to my primary work discussed here, during the term of the grant I published an article in *Physics Letters* outlining a new technique for computing bifurcations of dynamical systems [13].

3.1 Stochastic Learning

Since the original submission we have focused our work on stochastic learning in two areas. The first is directed at overcoming the limitations of the diffusion approximation. Towards this end we developed a perturbation expansion for solutions of the Kramers-Moyal equation. Our perturbation expansion provides the probability density as a power series in the learning rate μ. Though independently developed, the perturbation expansion is intimately linked to Van Kampen's system size expansion [9].

The second area focuses on asymptotic (late-time) behavior of algorithms with learning rate schedules $\mu(t) = \mu_0/t$. We are developing algorithms that use an *adaptive momentum parameter* help achieve nearly optimal convergence rate. Our algorithm improves on previous methods as it does not require estimates of the eigenvalue spectrum of the Hessian, as does Fabian's [14] approach; neither does the algorithm rely on measuring an auxiliary statistic, as does the approach developed by Darken and Moody [15].

Perturbation Expansion

In [2] we gave an equilibrium solution for the LMS algorithm obtained in the diffusion approximation, and showed its validity for small learning rates. In order to obtain more accurate equilibrium solutions, we have developed a perturbative expansion of solutions to the Kramers-Moyal equation[1] The latter contains all the dynamics of the probability density (we refer the reader to the original proposal).

Perturbation techniques, familiar from classical and quantum physics, enable one to construct approximate solutions to intractable problems that are similar to problems that can be solved in closed form. Here we will develop a perturbation expansion for solutions to the forward Kramers-Moyal equation. We limit our discussion to the equilibrium density for the LMS algorithm, though the technique extends to other problems, and to transient phenomena as well.

Stationary densities for the Kramers-Moyal equations are solutions to

$$\sum_{i=1}^{\infty} \frac{(-\mu)^i}{i!} \sum_{j_1 \ldots j_i = 1}^{m} \frac{\partial^i}{\partial \omega_{j_1} \partial \omega_{j_2} \ldots \partial \omega_{j_i}} \{ \langle H_{j_1} H_{j_2} \ldots H_{j_i} \rangle_x \, P_s(\omega) \} \equiv L_{KM} \; P_s(\omega) = 0 \qquad (1)$$

[1] This work is included in a long journal manuscript under preparation.

This is not solvable in closed form. The idea behind the perturbation technique is to write both the operator L_{KM} and the stationary solution P_s as a power series in a small parameter, and then to solve the resulting expression order-by-order in this parameter. Here, the naturally occurring perturbation parameter is the learning rate μ.

Schematically, the method proceeds as follows. We rewrite the operator L_{KM} as

$$L_{KM} \;=\; \mu\,(\,L_0 \,+\, \mu\,L_1 \,+\, \mu^2\,L_2 \,+\, \ldots)\;, \tag{2}$$

where the L_i will be made explicit below. Next we expand P_s as

$$P_s \;=\; P^{(0)} \,+\, \mu\,P^{(1)} \,+\, \mu^2\,P^{(2)} \,+\, \ldots \tag{3}$$

where the $P^{(i)}$ are to be determined. Now, we substitute (2) and (3) into (1) to obtain

$$\begin{aligned}&\mu\,L_0\,P^{(0)} \,+\\ &\mu^2\,(\,L_1\,P^{(0)} \,+\, L_0\,P^{(1)}\,) \,+\\ &\mu^3\,(\,L_2\,P^{(0)} \,+\, L_1\,P^{(1)} \,+\, L_0\,P^{(2)}\,) \,+\, \ldots \;=\; 0\;.\end{aligned} \tag{4}$$

In order for (4) to hold for arbitrary μ, the coefficients of each power of μ must separately vanish. Thus we obtain the set of equations

$$\begin{aligned}L_0\,P^{(0)} &\;=\; 0\\ L_0\,P^{(1)} &\;=\; -L_1\,P^{(0)}\\ L_0\,P^{(2)} &\;=\; -(\,L_2 P^{(0)} \,+\, L_1\,P^{(1)}\,)\quad \text{etc}\end{aligned} \tag{5}$$

The strategy is to successively solve each equation in (5). The key is to obtain a representation for L_{KM} such that L_0 is an operator with a known complete set of eigenfunctions. Each of the $P^{(i)}$ is then expanded in terms of the eigenfunctions of L_0. $P^{(0)}$ is simply the kernel of L_0.

Let us assume that we have such a representation for L_0 with eigenvalues $-\lambda$ and eigenfunctions F_λ

$$L_0\,F_\lambda \;=\; -\lambda\,F_\lambda\;. \tag{6}$$

The adjoint of L_0, denoted $L_0^\dagger$, has eigenvalues $-\lambda$ and eigenfunctions G_λ

$$L_0^\dagger\,G_\lambda \;=\; -\lambda\,G_\lambda\;. \tag{7}$$

It is easy to show that the two sets of eigenfunctions can be chosen to be biorthogonal

$$\int d\omega\,G_{\lambda'}(\omega)\;\;F_\lambda(\omega) \;\equiv\; (\,G_{\lambda'}, F_\lambda\,) = \delta_{\lambda',\lambda}\;. \tag{8}$$

Finally, given these basis eigenfunctions, the $P^{(i)}$ that satisfy (5) can be expanded as

$$P^{(0)} = F_0 \tag{9}$$

$$P^{(1)} = \sum_{\lambda \neq 0} \frac{1}{\lambda} \left(G_\lambda, L_1 P^{(0)} \right) F_\lambda \tag{10}$$

$$P^{(2)} = \sum_{\lambda \neq 0} \frac{1}{\lambda} \left\{ \left(G_\lambda, L_2 P^{(0)} \right) + \left(G_\lambda, L_1 P^{(1)} \right) \right\} F_\lambda \quad \text{etc.} \tag{11}$$

having used eigenequations (6) and the bi-orthogonality relation (8).

We have applied the technique outlined above to compute equilibrium densities for the LMS algorithm with targets generated by a noisy teacher neuron. We denote the displacement from the optimal weight ω_* by $v \equiv \omega - \omega_*$. It is convenient to make the change of variables

$$v = y \sqrt{\mu \sigma^2} \tag{12}$$

where μ is the learning rate and σ^2 is the variance of the teacher noise.

In the y coordinates, the first two operators in (2) are

$$\begin{aligned} L_0 &= R \left(\partial_y y + \frac{1}{2} \partial_y^2 \right) \\ L_1 &= \frac{3}{8} R^2 \left(4\, \partial_y^2 y^2 + 4\, \partial_y^3 y + \partial_y^4 \right) \end{aligned}$$

where R is the input correlation. Note that L_0 corresponds to an Ornstein-Uhlenbeck process. Its eigenfunctions are Gaussians multiplied by Hermite polynomials. The first two $P^{(i)}$ are

$$\begin{aligned} P^{(0)} &= \frac{1}{\sqrt{\pi}} e^{-y^2} \\ P^{(1)} &= \frac{3}{4} R \left(-1 + 2y^2 \right) P^{(0)} \ . \end{aligned}$$

A graphical comparison shows that the perturbative solutions are superior to those obtained from the Fokker-Planck equation. In Figure 1 we compare the Fokker-Planck, perturbative and experimentally derived densities for 1-D LMS.

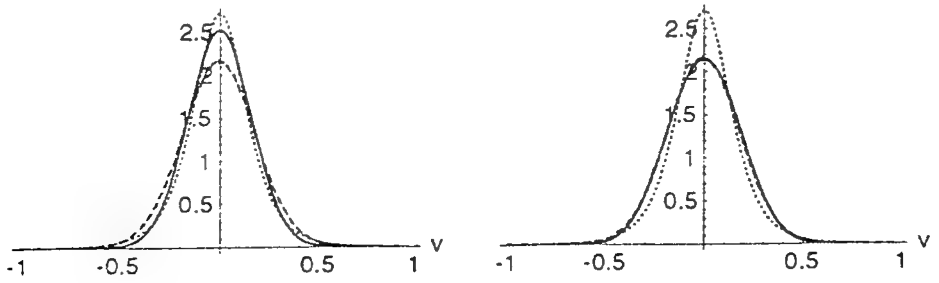


Figure 1: Simulated (dashed), Fokker-Planck (dotted), and perturbative (solid) densities for 1-D LMS with gaussian inputs ($\mu = 0.05$, $\sigma^2 = 1$, and $R = 4$). a) 0^{th}-order: $P^{(0)}$, b) 1^{st}-order: $P^{(0)} + \mu P^{(1)}$.

The perturbation solutions have been carried to higher order and applied to multidimensional problems as well.

Our perturbation technique is intimately linked to Van Kampen's system size expansion [9] for the Kramers-Moyal equation. To apply Van Kampen's expansion to neural net learning, one writes the weight error v as the sum of deterministic plus fluctuation pieces

$$v = \phi + \sqrt{\mu}\, y \ . \tag{13}$$

The deterministic piece ϕ evolves by descent along the average or true gradient, approaching zero exponentially. At late times, only the dynamics of the fluctuations y remain. These are described to lowest order by L_0 (and this is what Van Kampen treats). Our perturbation expansion extends the description to arbitrary order.

Asymptotics and Optimal Convergence

Learning rate schedules of the form $\mu = \mu_0/t$ give rise to weight vector sequences that converge in mean square to local optima ω_*. The asymptotic rate of convergence is conveniently characterized by the expected squared weight error (or misadjustment) $E\left[|v|^2\right] \equiv E\left[|\omega - \omega_*|^2\right]$. It is well known [15, 16, and references therein] that the convergence rate depends on whether μ_0 is larger than the critical value

$$\mu_c = \frac{1}{2\,\lambda_{min}} \tag{14}$$

where λ_{min} is the smallest eigenvalue of the cost function's Hessian evaluated at ω_*. If $\mu_0 > \mu_c$ then the misadjustment falls off asymptotically as $1/t$, whereas if $\mu_0 < \mu_c$ the misadjustment falls off *slower* than $1/t$.

To achieve the optimal rate, one must estimate μ_c and adjust μ_0 to be larger ($\mu_0 = 1/\lambda_{min}$ is optimal). Fabian [14] estimates the Hessian during the optimization, and uses the estimate to readjust μ_0. This is clearly not feasible for high-dimensional optimization problems. Darken and Moody [15] measure a statistic that characterizes the roughness of trajectories, and use the time evolution of this statistic to adjust μ_0. This approach, though less storage intensive than estimating the Hessian's eigenvalue spectrum, requires computation not central to the search process.

We are proposing an alternative solution based on stochastic gradient descent with momentum. Using the Kramers-Moyal expansion we have rederived the classic results on convergence rates (and related results on asymptotic normality) and extended them to stochastic gradient descent with momentum [4, 3]. The analysis shows that at late times, learning is governed by an *effective* learning rate

$$\mu_{eff} \equiv \frac{\mu}{1 - \beta} \tag{15}$$

where β is the coefficient of the momentum term.

Adaptive Momentum Improves Convergence

Based on our work on asymptotic convergence with momentum, we have devised an algorithm that forcest the expected squared misadjustment $E[|v|^2]$ to fall off nearly as $1/t$ at late times. The new algorithm does not involve estimating Hessian and its eigenspectrum, nor does it involve calculating an auxiliary statistic, nor does it involve setting μ_0. Instead the momentum parameter β is adapted on-line.

For simplicity, consider momentum gradient descent in 1-D. Based on the critical value for the learning rate (14) and the effective learning rate with momentum (15), it is clear that if the momentum coefficient were *set to* $\beta = 1 - \mu_0 R$, (R the Hessian) then we would achieve the optimal convergence rate $E[|\omega - \omega_*|^2] \propto 1/t$.

Of course we do not know R, but it can be estimated on-line. For LMS, $R = E[X X^T]$ and a convenient estimate is $\hat{R} = X_t X_t^T$. For bounded inputs, one can show that an algorithm based on this choice achieves the optimal $1/t$ convergence rate.

Figure 2 shows the results of our adaptive momentum algorithm on a 2-D LMS problem. The plots show, on a log-log scale, the expected squared misadjustment (computed from an ensemble of 1000 networks) as a function of time. Optimal convergence with $E[|v|^2] \propto 1/t$ corresponds to slope -1 on these curves. The correlation eigenvalues for the input data are $\lambda_1 = 0.4$, $\lambda_2 = 4.0$, so $\mu_c = 1.25$. The plot on the left shows the convergence of *standard* LMS with learning rate μ_0/t for the three initial rates $\mu_0 = (1.5, 1.0, 0.25)$. Only the curve corresponding to $\mu_0 = 1.5$ exhibits the optimal convergence rate. The plot on the right shows the results with adaptive momentum. Notice that regardless of μ_0, at late times all the curves exhibit the optimal convergence rate.

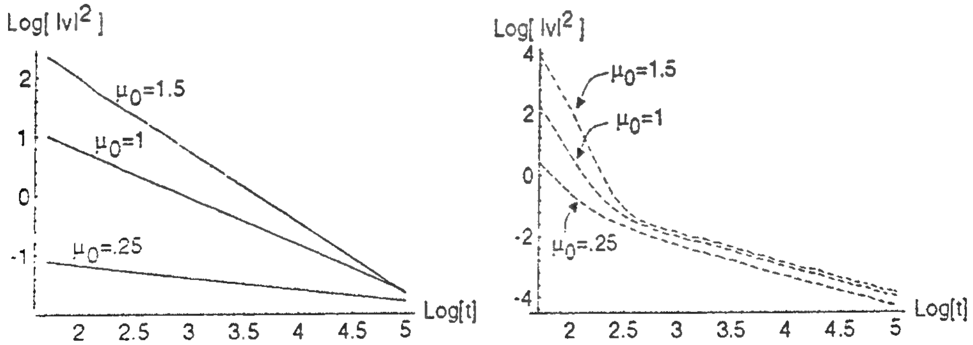


Figure 2 Expected squared misadjustment for an ensemble of 1000 LMS networks. Correlation eigenvalues $\lambda_1 = 0.4$, $\lambda_2 = 4.0$. LEFT - no momentum. RIGHT - adaptive momentum.

Similar results are obtained for problems with larger condition numbers. Figure 3 shows simulation results without momentum, and with adaptive momentum for condition number of $\approx 10^4$. In this simulation, the annealing and adaptive momentum are started at late times. Without momentum, the convergence is stalled - at late times the slove of the error curve is essentially zero. With adaptive momentum, the asymptotic slope is ≈ -0.66. Although this is not the theoretical optimal value (slope −1), the improvement in convergence, relative to to no momentum is substantial.

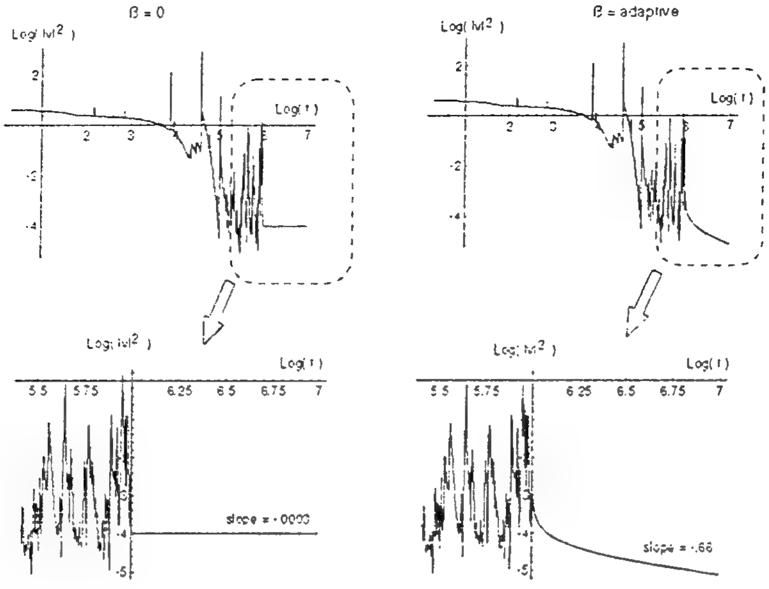


Figure 3 Simulation of 4-D LMS for condition number $\lambda_{max}/\lambda_{min} \approx 10^4$.

Further work on these algorithms integrates automatic passage from constant learning rate to annealing. We are currently extending the algorithm to non-linear optimization problems and will start executing it on large speech-recognition networks.

3.2 Non-Linear Dimension Reduction

This work is aimed at developing algorithms for data modeling and compression that perform better than linear techniques, and are fast to train. Initially we have approached the problem of data dimension reduction. Kramer [17], DeMers [18], and Oja [19] have proposed the use of 5-layer feed-forward auto-associative networks with a bottle-neck middle layer to perform non-linear dimension reduction. These networks have input and output layers of size equal to the dimension of the data. There are three layers of hidden nodes. The number of nodes in the second hidden layer is equal to the dimension of the encoded signal. The networks are trained to perform an identity transformation on the input data. After training, the low-dimensional encoding is extracted from the activities of the nodes in the second hidden layers. These networks are able to provide more accurate encodings than the principal component analysis (PCA). However they are slow to train and are prone to trapping in poor solutions.

As described in the original proposal, we have developed an algorithm that uses *local* PCA to reduce data dimension. The algorithm partitions the space using a vector quantizer (VQ) and then performs a PCA projection within each of the Voronoi cells of the partition. The PCA captures the local structure of the data, while the distribution of Voronoi cells captures the global, non-linear structure of the data.

We originally exercised the algorithm on speech data and compared its performance with global PCA, and with the global non-linear model implemented by a 5-layer auto-associative network [20]. The original data consists of 32 DFT coefficients (spanning the frequency range 0-4kHz) of the monothongal vowels extracted from continuous speech. The goal was to reduce the data to a low (two- or three-dimensional) representation. The figure of merit for these experiments was the error incurred in reconstructing the original signal from the dimension-reduced representation.

Both the neural network and the local-linear technique provided lower reconstruction error than PCA. The local linear technique provided roughly a 40% decrease in error relative to the neural net, and trained up to an order of magnitude faster [20].

Under the present grant, we exercised the algorithm on image data, again comparing its performance with a neural network, and with PCA [21]. In those experiments, we used the image database developed by DeMers and Cottrell, comparing our results with their study of dimension reduction with neural networks [22]. The original 64x64 images were first encoded into 50 principal components. This 50-dimensional representation was then reduced to 5 dimensions using either of the nonlinear techniques. A linear reduction to 5 dimensions was provided by retaining only the leading 5 principal components. As with the experiments on speech data both non-linear techniques outperformed PCA. Our locally linear algorithm outperformed the neural network [21]. For example, one of the neural network models achieved a normalized reconstruction error[2] of 0.07 and required 31,980 cpu seconds to train [3]. A comparable locally linear model (normalized error 0.0696) required only 50 cpu seconds to train.

Sample results of the image reconstruction are shown in figure 3 The images clearly portray both the superiority of the non-linear techniques over PCA, and the superiority of our locally linear technique over the neural network (mouth shape is especially revealing in this series).

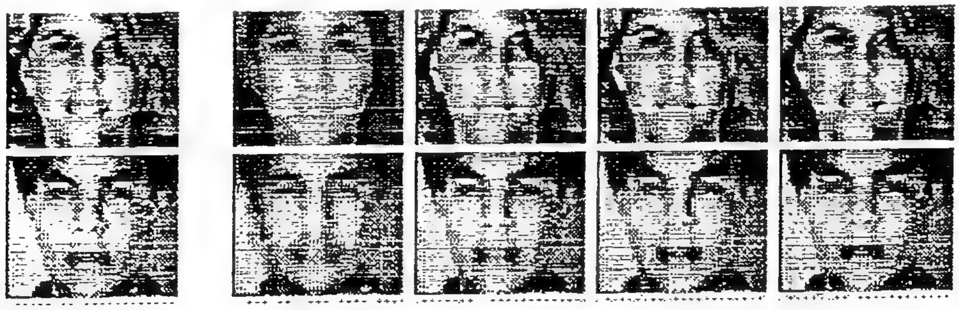


Figure 3: Two representative images: Left to right – Original 50-PC image, reconstruction from 5-D encodings: PCA, Best neural network model, locally linear model with 10 partitions, locally linear model with 50 partitions.

[2] The error measure is the mean squared reconstruction error divided by the mean squared signal strength.

[3] Using a five-layer autoassociative network, DeMers and Cottrell [22] obtain an normalized reconstruction error of 0.1317 for the same data.This is comparable to our results.

Coupling the Partition to the Projection

The algorithm discussed above, though very effective, is not optimal. The algorithm first partitions the input space into disjoint Voronoi cells using a standard VQ algorithm (e.g. LBG[4] or competitive learning). After this partition is built, the local PCA projection is performed within each cell of the partition. Thus the initial partitioning is independent of the projection that follows.

One would expect to obtain lower distortion in the final representation if the partition was built in a manner that was determined in part by the projection. Indeed the two steps can be coupled by building the partition so as to *minimize the reconstruction error after the PCA projection*. Either gradient descent or LBG-like algorithms can be built using this distortion measure. We have coded such algorithms and initial experiments indicate a 10-20% reduction in error relative to the original algorithm in which the partition is constructed independently of the projection. More detail is available in [5].

3.3 Network Pruning

Fitting model complexity to data is one of the outstanding problems in statistical model building. Rich models, those with many parameters, allow close fits to sample data, but may perform badly on out-of-sample data (so-called generalization performance). To counter this over-fitting, various techniques have been introduced that decrease model variance at the expense of model bias in order to improve performance on out-of-sample data.

- *Regularization* schemes (e.g. weight decay) add a penalty term to the cost function. The proper coefficient for this term is not known a priori, so one must perform several optimizations with different values; a cumbersome process.
- *Weight-elimination* schemes (e.g. optimal brain damage [24] and its derivative optimal brain surgery [25]) involve traning large nets and then removing the weights that least affect the training error. These techniques require calculating the Hessian or some approximation to it. Calculating the full Hessian is impractical for large nets, and the approximations are often poor.
- *Early stopping* monitors the error on a validation set and halts learning when this error starts to increase.

We have developed an alternative technique that uses principal component analysis (PCA) in conjunction with supervised learning. Briefly stated, the technique uses PCA to decorrelate node activities and then eliminates the (decorrelated) degrees of freedom that have the least effect on the output error (least bias). The technique is fast to implement and achieves good results on both linear and non-linear models. Our paper, delivered as an oral presentation at

[4] The Linde-Buzo-Gray algorithm [23, and references therein] is the commonly-used batch-mode algorithm for designing a vector quantizer.

the 1993 NIPS meeting, gives more details of the algorithm implementation and results on several sample problems.

3.4 Learning Invariances

In machine learning one sometimes wants to incorporate invariances into the function learned. Our knowledge of the problem dictates that the machine outputs ought to remain constant when its inputs are transformed under a set of operations $\mathcal{G}$[5]. In character recognition, for example, we want the outputs to be invariant under shifts and small rotations of the input image.

There are several ways to achieve this invariance

1. The invariance can be built into the input representation. In image processing the use of Fourier amplitude coefficients, rather than pixel intensities, provides invariance under translations.
2. In neural networks, the invariance can be hard-wired by weight sharing in the case of summation nodes [26] or by constraints similar to weight sharing in higher-order nodes [27].
3. One can enhance the training ensemble by adding examples of inputs transformed under the desired invariance group, while maintaining the same targets as for the raw data.
4. One can add to the cost function a regularizer that penalizes changes in the output when the input is transformed by elements of the group [10, 11].

Intuitively one expects the approaches in 3 and 4 to be intimately linked.

This link is established by writing the probability distribution for the enhanced training set in terms of the original distribution and the distortions introduced. These transformations, or distortions, of the inputs are carried out by group elements $g \in \mathcal{G}$. For Lie groups[6], the transformations are analytic functions of parameters $\alpha \in R^k$

$$x \rightarrow x' = g(x;\alpha) \ , \tag{16}$$

with the identity transformation corresponding to parameter value zero

$$g(x;0) = x \ . \tag{17}$$

By adding distorted input examples we alter the original density $p(x)$. We characterize the frequency with which different distortions are represented in the enhanced ensemble by a probability density over group parameters $p(\alpha)$. With this density, the distribution for the distortion-enhanced input ensemble becomes

$$\begin{aligned} p(x') &= \int\int d\alpha\, dx\; p(x'|x,\alpha)\; p(\alpha)\; p(x) \\ &= \int\int d\alpha\, dx\; \delta(\, x' - g(x;\alpha)\,)\; p(\alpha)\; p(x) \ , \end{aligned}$$

[5] We assume that the set forms a group.

[6] See for example [28].

where $\delta(\cdot)$ is the Dirac delta function.

Finally we impose that the targets remain unchanged when the inputs are transformed according to (16) *i.e.*, $p(t|x') = p(t|x)$.

In this framework, the cost function for the distortion-enhanced training data is shown to be equivalent to the cost function for the original (untransformed) data, plus a regularizer term.

For unbiased models, the regularizer is shown to reduce to a simple penalty for violation of the desired invariance:

$$\mathcal{E}_R \equiv \int d\alpha\; p(\alpha) \int dx\;\; p(x)\;\; [\, f(x, w) - f(g(x;\alpha); w)\,]^2 \tag{18}$$

Our publications [8] contain further detail.

References

[1] Genevieve B. Orr and Todd K. Leen. Weight space probability densities in stochastic learning. II. Transients and basin hopping times. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[2] Todd K. Leen and John E. Moody. Weight space probability densities in stochastic learning: I. Dynamics and equilibria. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[3] Todd K. Leen and Genevieve B. Orr. Momentum and optimal stochastic search. In M.C. Mozer, P. Smolensky, D.S. Touretzky, J.L. Elman, and A.S. Weigend, editors, *Proceedings of the 1993 Connectionist Models Summer School*, Hillsdale, NJ, 1993. Erlbaum Associates.

[4] Todd K. Leen and Genevieve B. Orr. Optimal stochastic search and adaptive momentum. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, San Francisco, CA., 1994. Morgan Kaufmann Publishers.

[5] Todd K. Leen and Nanda Kambhatla. Fast non-linear dimension reduction. In *Advances in Neural Information Processing Systems, Natural and Synthetic*. Morgan Kauffmann, Feb 1994.

[6] Nanda Kambhatla and Todd K. Leen. Classifying with gaussian mixtures, clusters, and subspaces. To appear in Advances in Neural Information Processing Systems, 7, 1994.

[7] Asriel Levin. Todd K. Leen, and John E. Moody. Fast pruning using principal components. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*. San Francisco. CA., 1994. Morgan Kaufmann Publishers.

[8] Todd K. Leen. From data distributions to regularization in invariant learning. To appear in Neural Computation and Advances in Neural Information Processing Systems 7, 1994.

[9] N. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, Amsterdam, 1981.

[10] Patrice Simard, Bernard Victorri, Yann Le Cun, and John Denker. Tanget prop - a formalism for specifying selected invariances in an adaptive network. In John E. Moody, Steven J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 895–903. Morgan Kaufmann, 1992.

[11] Yasar S. Abu-Mostafa. A method for learning from hints. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems, vol. 5*, pages 73–80. Morgan Kaufmann, 1993.

[12] Asriel Levin and Todd K. Leen. Using PCA to improve generalization in supervised learning. Talk delivered at the NATO Workshop on Statistics and Neural Nets, Les Arcs, France, June 1993.

[13] Todd K. Leen. A coordinate-independent center manifold reduction. *Physics Letters*, A-, 1993.

[14] V. Fabian. Asymptotically efficient stochastic approximation; the RM case. *The Annals of Statistics*, 1:486–495, 1973.

[15] Christian Darken and John Moody. Towards faster stochastic gradient search. In J.E. Moody, S.J. Hanson, and R.P. Lipmann, editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[16] Larry Goldstein. Mean square optimality in the continuous time Robbins Monro procedure. Technical Report DRB-306, Dept. of Mathematics, University of Southern California, LA, 1987.

[17] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37:233–243, 1991.

[18] David DeMers. Dimensionality reduction for non-linear time series. In *SPIE Conference on Neural & Stochastic Methods for Image and Signal Processing*, 1992.

[19] E. Oja. Data compression, feature extraction, and autoassociation in feedforward neural networks. In *Artificial Neural Networks*, pages 737–745. Elsevier Science Publishers B.V. (North-Holland), 1991.

[20] Nandakishore Kambhatla and Todd K. Leen. Fast non-linear dimension reduction. In *IEEE international Conference on Neural Networks, Vol. 3*, pages 1213–1218. IEEE, 1993.

[21] Nanda Kambhatla and Todd K. Leen. Fast non-linear dimension reduction. In J.D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, San Francisco, CA., 1994. Morgan Kaufmann Publishers.

[22] David DeMers and Garrison Cottrell. Non-Linear dimensionality reduction. In Giles, Hanson, and Cowan, editors, *Advances in Neural Information Processing Systems, vol. 5*, San Mateo, CA, 1993. Morgan Kaufmann.

[23] Robert M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, April 1984.

[24] John S. Denker Yann Le Cun and Sara A. Solla. Optimal brain damage. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. Morgan Kaufmann Publishers, 1990.

[25] Babak Hassibi, David G. Stork, and Gregory J. Wolff. Optimal brain surgeon and general network pruning. Technical Report CRC-TR-9235, RICOH California Research Center, Menlo Park, CA 94025, September 1992.

[26] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems, vol. 2*, pages 396–404. Morgan Kaufmann Publishers, 1990.

[27] C.L. Giles, R.D. Griffin, and T. Maxwell. Encoding geometric invariances in higher-order neural networks. In D.Z.Anderson, editor, *Neural Information Processing Systems*, pages 301–309. American Institute of Physics, 1988.

[28] D.H. Sattinger and O.L. Weaver. *Lie Groups and Algebras with Applications to Physics, Geometry and Mechanics*. Springer-Verlag, 1986.